Министерство образования и молодежной политики Свердловской области

ГАПОУ СО «Екатеринбургский колледж транспортного строительства»

Отчёт по программе  «Практическое занятие 12»

Выполнил: Рудных Екатерина

Группа: ПР-21

Преподаватель: Мирошниченко Г.В

2025

**Задание 1:**

1.Реализовать через корутины вызов функций нажатий и

параллельного получения данных из n функций (n вводит пользователь).

Затем, нужно их объединять и отображать.

**Входные данные:**

1. Количество задач (m)

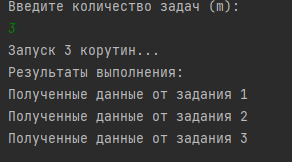
**Выходные данные:**

1. Сообщения о ходе выполнения
2. Результаты выполнения корутин

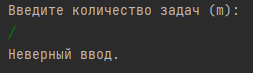
**Листинг программы:**

import kotlinx.coroutines.\*  
import java.util.\*  
suspend fun getData(taskIndex: Int): String {  
 delay(800L)  
 return "Полученные данные от задания $taskIndex"  
}  
fun main() {  
 *println*("Введите количество задач (m):")  
 val userInput = *readLine*()  
 if (userInput == null || userInput.*isEmpty*() || userInput.*toIntOrNull*() == null || userInput.*toInt*() <= 0) {  
 *println*("Неверный ввод.")  
 return  
 }  
 val taskCount = userInput.*toInt*()  
 *println*("Запуск $taskCount корутин...")  
 val outputData = Array(taskCount) **{** "" **}** *runBlocking* **{** for (index in 1..taskCount) {  
 *launch* **{** val data = getData(index)  
 outputData[index - 1] = data  
 **}** }  
 **}** *println*("Результаты выполнения:")  
 for (data in outputData) {  
 *println*(data)  
 }  
}

**Вывод:**

****

**Тестовые ситуации:**



**Задание 2:**

Создайте программу для загрузки участников для всех репозиториев

данной организации, с сортировкой пользователей по количеству их

репрозиториев.

Вводится имя пользователя и токен (или пароль) GitHub в соответствующие

поля. Если в раскрывающемся меню вариантов выбран вариант

«БЛОКИРОВКА», нажмите «Загрузить участников». Пользовательский

интерфейс должен зависнуть на некоторое время, а затем отобразить

отсортированный список участников с указанием количества репрозиториев .

**Входные данные:**

1. Имя организации
2. Количество репозиториев
3. Владельцы репозиториев

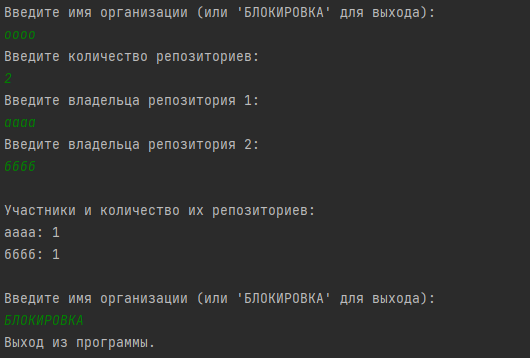
**Выходные данные:**

1. Сообщения о ходе выполнения
2. Результаты

**Листинг программы:**

import kotlinx.coroutines.\*  
import kotlinx.coroutines.flow.\*  
  
fun main() = *runBlocking* **{** while (true) {  
 *println*("Введите имя организации (или 'БЛОКИРОВКА' для выхода):")  
 val orgName = *readLine*() ?: break  
 if (orgName.*uppercase*() == "БЛОКИРОВКА") break  
  
 *println*("Введите количество репозиториев:")  
 val repoCount = *readLine*()?.*toIntOrNull*() ?: 0  
 val repos = *List*(repoCount) **{** index **->** *println*("Введите владельца репозитория ${index + 1}:")  
 val owner = *readLine*() ?: "Неизвестный"  
 Repo(Owner(owner))  
 **}** val counts = repos.*groupingBy* **{ it**.owner.login **}** .*eachCount*()  
 .*toList*()  
 .*sortedByDescending* **{ it**.second **}** *println*("\nУчастники и количество их репозиториев:")  
 counts.*forEach* **{** (owner, count) **->** *println*("$owner: $count") **}** *println*()  
 }  
 *println*("Выход из программы.")  
**}**data class Repo(val owner: Owner)  
data class Owner(val login: String)

**Вывод:**

****

**Тестовые ситуации:**

